EXAMINER'S AMENDMENT

1. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Mr. Jacob Rohwer (Reg. No. 61,229) on February 19, 2009.

2. The application has been amended as follows:

Claim 1. (Currently Amended) A software architecture embodied on one or more computer-readable storage media, the software architecture executed by a computing device for a distributed computing system comprising:

an application configured to handle requests submitted by remote devices over a network; and

a multi-tiered framework comprising:

an application program interface layer organized into multiple root namespaces, the application program interface layer to present functions used by the application to access network and computing resources of the distributed computing system;

a common language runtime layer, wherein calls to the application program interface layer are handed to the common language runtime layer supporting applications written in a plurality of different languages and translated into an intermediate supported language, the application program interface layer comprising various types related to constructing user interfaces, wherein the types define a collection of classes, interfaces, delegates, enumerations, and structures which belong to a group assigned a group name associated with one of the root namespaces, and wherein each of the types is referenced by a hierarchical name comprising a top level identifier prefixed to the group name; ~~and~~

an operating system layer or an object model service, wherein the calls handed to the common language runtime layer are executed locally by the operating system layer or the object model service; and

a common language specification for local execution by the operating system layer or the object model service, wherein the common language specification provides an ability to use a particular code module written in a first programming language with a code module written in a second programming language.

Claim 5. (Currently Amended) A multi-tiered architecture including an application program interface layer embodied on one or more computer readable storage media, comprising:

multiple types related to constructing user interfaces;

individual types being associated with one or more groups and being referenced by one or more hierarchical names, wherein each hierarchical name includes a top level identifier prefixed to a group name assigned to one of the one or more groups, the types comprising:

classes which represent managed heap allocated data that has reference assignment semantics;

interfaces that define a contract that other types can implement;

delegates that are object oriented function pointers; and structures that represent static allocated data that has value assignment semantics and enumerations which are value types that represent named constants;

wherein the application program interface layer is associated with:

a common language runtime layer supporting applications written in a plurality of different languages and translated into an intermediate language supported by the common runtime layer; and

a common language specification for local execution by an operating system or an object model service, wherein the common language ~~runtime~~ specification provides the ability to use a particular code module written in a first programming language with a code module written in a second programming language.

Claim 16. (Currently Amended) A distributed computer software architecture embodied on one or more computer-readable storage media, the distributed computer software architecture comprising:

one or more applications configured to be executed on one or more computing devices, the applications handling requests submitted from remote computing devices;

a networking platform to support the one or more applications;

an application programming interface to interface the one or more applications with the networking platform, the application programming interface comprising various types related to constructing user interfaces, individual types being associated with one or more groups and being referenced by one or more hierarchical names, wherein each of the hierarchical names includes a top level identifier prefixed to a group name assigned to one of the one or more groups; ~~and~~

a common language runtime layer supporting applications written in ~~one or more different languages~~ a plurality of different languages and translated into an intermediate language supported by the common runtime layer and a common language specification for local execution by an operating system or an object model service; and

a common language specification for local execution by the operating system or the object model service, wherein the common language specification provides an ability to use a particular code module written in a first programming language with a code module written in a second programming language.

Claim 28. (Currently Amended) A computer system including one or more microprocessors and one or more software programs stored on one or more computer-readable storage media, the one or more software programs utilizing an application program interface to request services from an operating system, the application program interface including separate commands to request services comprising services related to constructing user interfaces, wherein the application program interface groups API functions into multiple namespaces that define a collection of classes which represent managed heap allocated data that has reference assignment semantics, interfaces that define a contract that other types can implement, delegates that are object oriented function pointers, enumerations which are value types that represent named constants and structures that represent static allocated data that has value assignment semantics, the application program interface being associated with:

a common language runtime layer supporting applications written in ~~one or more different languages~~ a plurality of different languages and translated into an intermediate language supported by the common runtime layer; and[[,]]

a common language specification for local execution by an operating system or an object model service, wherein the common language specification provides an ability to use a particular code module written in a first programming language with a code module written in a second programming language; and

wherein the type comprising the interfaces comprises one or more of the following interfaces:

a button control interface that allows a control to act like a button on a form;

a container control interface that provides functionality for a control to act as a parent for other controls;

an editing notification interface;

a data object interface that provides a format independent mechanism for transferring data;

a feature support interface that specifies a standard interface for retrieving feature information from a current system;

a message filter interface; or

a handle-exposing interface to expose handles.

Claim 29. (Currently Amended) A method, comprising:

managing network and computing resources for a distributed computing system;

exposing a set of functions that enable developers to access the network and computing resources of the distributed computing system, the set of functions comprising functions to facilitate construction of user interfaces, wherein the user interfaces include windowing, menus, and dialogs, and wherein the functions are grouped into multiple namespaces that define a collection of classes which represent managed heap allocated data that has reference assignment semantics, interfaces that define a contract that other types can implement, delegates that are object oriented function pointers, enumerations which are value types that represent named constants and structures that represent static allocated data that has value assignment semantics; ~~and~~

using a common language runtime layer supporting applications written in ~~one or more different languages~~ a plurality of different languages and translated into an intermediate language supported by the common runtime layer; and

using a common language specification for local execution by an operating system or an object model service, wherein the common language specification provides an ability to use a particular code module written in a first programming language with a code module written in a second programming language.

Claim 31. (Currently Canceled)

Claims 33-40. (Currently Canceled)

3. The drawings filed 07/10/2001 are acceptable.

4. The following is an examiner's statement of reasons for allowance:

As to 1, 3-16, 18-30, 41 and 42, the prior art of record does not teach or render obvious the limitations recited in claims 1, 5, 16, 28 and 29, when taken in the context of the claims as a whole, specific to an application program interface layer comprising various types relating to constructing user interfaces, a common language runtime layer supporting applications written in a plurality of different languages and translated into an intermediate language supported by the common runtime layer and a common language specification for local execution by an operating system or an object model service, wherein the common language ~~runtime~~ specification provides the ability to use a particular code module written in a first programming language with a code module written in a second programming language.

Moreover, evidence for modifying the prior art teachings by one of ordinary skill level in the art was not uncovered so as to result in the invention as recited in claims 1, 5, 16, 28 and 29.

Especially, the prior art of record is overcome by the Affidavit filed 12/9/2008.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

5. Any inquiry concerning this communication or earlier communications from the examiner should be directed to DIEM K. CAO whose telephone number is (571)272-3760. The examiner can normally be reached on Monday - Friday, 7:30AM - 4:30PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Meng-Ai An/
Supervisory Patent Examiner, Art Unit 2195

DC
February 22, 2009.